Risk Reduction Demonstration Pilot

IV&V Workshop

September 13-15, 2011

Gary Barber

The Problem

- IV&V is generally seen as a risk reduction technique
- Finding issues and ensuring their resolution surely reduces overall risk, but:
  - In what way?
  - By how much?
  - What risk was reduced?
  - What activity reduced the risk?
- IV&V can have impact in multiple ways
  - IV&V can perform analyses to show that the developer is correctly and completely developing requirements, design, implementation and verification of risk-related items
  - IV&V can bring problems and deficiencies to the attention of the developer who will then take action to correct the situation
  - IV&V can perform activities independent of the developer (who may be time and/or resource constrained) that serve to mitigate risk
- A method to demonstrate risk reduction would provide another important means to communicate IV&V value to our customer
- A structured means of identifying risk and defining risk contributors per risk would help to ensure we are performing the most productive activities, i.e. the things that most significantly reduce the most critical risks

Proposed Method

- We need a method that demonstrates risk reduction resulting from IV&V activities
  (1) Develop a structured method to identify the risks that exist in a project
  (2) Develop a codified method to measure risk reduction via IV&V analyses and evidence collection
- Part 1 - Risk Identification
  - Need a structured and formalized method to provide assurance that all important risks have been identified
    - Comprehensive and reusable risk catalog
    - Include a variety of aspects of space software development, IV&V historical risk data, and available literature
  - Need a rigorous approach for using the catalog to identify risks for a specific project
    - Develop a reasoning path method for a hierarchical thread of topical areas to examine for risk exposure
    - Include project specific information such as hazard analyses and fault trees, unique technologies, fault protection schemes, testing capabilities
- Part 2 - Risk Reduction Measurement
  - Use assurance case method as a model for risk tree development
  - Build a tree that identifies the risk decomposition
  - Identify criteria that would provide confidence that risk contribution from each branch has been mitigated (helps focus IV&V activities on high criticality areas)
  - Collect evidence that satisfies the criteria
  - Score the evidence
  - Roll up the branch scores to define the risk level
  - Assess risk reduction from scoring evidence assessment at various points in lifecycle

Simple Example of Risk and Claims

- There is risk that the Service Module fairing jettison operation will contain a defect that leads to mission failure
  1. There is risk that the software triggers to initiate the event will not operate correctly
  2. There is risk that the software actions to execute the process will not operate correctly
  3. There is risk that the software mechanisms to confirm the event occurred will not operate correctly
  4. There is risk that the software guards to prevent inadvertent activation of the event will not operate correctly
  5. There is risk that the steps to take if the event does not occur will not operate correctly

Example (cont)
Sub-claims to Claim 1

1 There is risk that the software triggers to initiate the event will not operate correctly

- 1a. There is risk that the software will not detect second stage ignition properly
- 1b. There is risk that the software will not confirm the passage of 25 seconds after second stage ignition
- 1c. There is risk that the SM fairing will not be jettisoned sufficiently before the LAS is jettisoned to prevent re-contact due to plume turbulence
- 1d. There is risk that confirmation of vehicle attitude rates need to be within constraints of safe jettison of SM fairing panels also needs to be a condition on the trigger

Example (cont)
Evidence Criteria for Subclaim 1a

1a. There is risk that the software will not detect second stage ignition properly
  - There are correct and complete requirements, design, implementation and test to detect second stage ignition
  - There are redundant and independent sources of the signal to indicate second stage ignition
  - Signal sources are based on physical evidence (e.g., unmistakable acceleration or unmistakable change of temperature in engine bell)
  - There is redundant and independent software to perform detection of a valid signal
  - A valid detection of second stage ignition is passed to the software starting the timer
  - This detection information cannot be lost due to single failure of communication path, software partition or task, or race condition of signals coming from redundant detection software
  - We look for redundancy here because failure of this event could cause mission failure and there are other requirements to withstand one failure for those kinds of events

Risk Reduction Demonstration Pilot

- Due to time constraints, the pilot focused only on part 2 (risk reduction) of the method
- Part 1 (risk identification) of the method will be examined at a later time
- Picked GRAIL as a typical science project nearing its completion
- Used two different risk types
  - Technical risk – Lunar Orbit Insertion
    - Very critical – if this does not happen correctly all mission objectives will be unmet
  - Process risk – conduct of Mission Readiness Review
    - Timely – this milestone was approaching at the time of the pilot
- Developed two separate techniques
  - Lightweight method to perform quick study of full set of risks identified and thin out the set for focus
  - Heavyweight method to study full depth and breadth of a critical risk

Lightweight Method

- Used to filter the potentially large set of risks identified near the start of a project
- Claims and sub-claims are accomplished only to the third or fourth level
- Each level of final decomposition is weighted and scored
- Weight is a subjective estimate of the level of importance
- The score is a subjective estimate of this part of the decomposition not working correctly
- Both the weight and the score are evaluated with red, yellow, or green
- The color for each node is rolled up to show a color for the top-level risk
- The final color at the top of the risk is used to sort the risks to find the most critical set

Heavyweight Method

- Develops a full description of the risk decomposition
- Subject Matter Expert (SME) develops branches using detailed knowledge to the extent known at the time of tree development
  - Each branch is decomposed to build a risk tree until the combination of extant design and SME knowledge provides no further decomposition
  - SME defines evidence criteria to give confidence the node will or will not leak one or more critical defects to operations
  - SME collects evidence related to criteria defined
  - SME evaluates evidence to provide a score related to confidence that critical defect leaks have or have not been mitigated
- This tree is enhanced as knowledge is gained through the project lifecycle

Build a Risk Tree

- Product design and SME knowledge was used to identify what contributes to or mitigates risk realization
  - For example:
    - Conditions that need to be true
    - States that product must be in
    - Design components that must function correctly
- Branches and nodes evolved as the analysts learned more about the system
- Developed two sub-trees, one for Hardware/Software interface and one for Software
- A spreadsheet was used to document the risk tree decomposition

Risk Tree Building
Lessons Learned

- A risk branch should terminate naturally rather than at the same level for all branches
- The risk claims should be restricted to software or software/hardware interface items
- It should be possible to reuse segments of the risk tree when analyzing other risks
- Although there are good graphical tools for drawing assurance cases, Excel has been used to develop the tree and perform the underlying calculations for scoring
- Application of Subject Matter Experts and thorough peer review are necessary to ensure a well-developed risk tree
- The generation of the risk tree is very dependent on the current stage of project development
- The risk tree will evolve as the project matures; active maintenance will be necessary to achieve maximum benefit
- Evidence should be applied to the current lowest level of sub-claim. However, not all evidence categories are necessary to satisfy each sub-claim.

Identify Evidence Criteria

- After making the last sub-claim for a given branch for a given round of evolution, the SME defines what would make him/her believe that the node will or will not contribute to critical defect leakage
- As the tree evolves, this set of criteria will become more detailed through further knowledge of the extant branches and nodes and further decomposition of the branches and nodes
- The criteria includes analyses of requirements, design, implementation, and verification artifacts

Collect Evidence

- Includes analyses of requirements, design, implementation, and verification to confirm completeness and correctness
- Focused on components involved with the node risk contributors
- The analysts perform the analysis defined by the evidence criteria and either:
  - Find one or more defects and confirm their correction
  - Find no defects and increase confidence in correctness of risk contributors
  - Perform other independent activities (e.g., independent test) that confirm completeness and correctness of risk contributors
- It can also happen that completion of analysis will result in belief of higher risk
  - IV&V finds a high severity risk that the developer will not fix
  - IV&V finds an unusually high number of defects
  - In these cases the risk score will be raised but annotated with reason

Score Evidence

- Evidence is scored at each node in the risk tree
- First define the importance of each node
  - Importance is based on level of contribution to risk realization
  - Level is defined in a table based on consequence and protection from (e.g., FDIR) a critical defect in a node
- Evidence is scored for each fundamental operational defect source – requirements, design, code
  - Use data from the Return On Investment study to define a range of defect densities that will leak to operations with and without IV&V (other data could be used here)
  - Use data from ORBIT to identify the percentage of defects found that are critical (severity 1 or 2) – found to be 3.6%
  - The SME evaluates the evidence collected and assigns a value between the top and bottom of the range that represents expected density of critical defects that will leak to operations
  - Evidence from verification analysis activities is applied to the scores for requirements, design, or code as appropriate
- The evidence score from each analysis type is added because a defect from any one could cause risk realization
- The confidence for the node is the importance times the sum of the evidence scores

Roll Up the Scores

- We use the Dempster-Shafer method to roll the scores up the branches of the tree
  - Provides a means for combining evidence
  - Gives a belief based on what we know so far
  - Industry standard
- Basic formula is

$$B(C) = 1 - (1 - (B(A))(1 - B(B))$$

  - This combines the belief in branches A and B to be the belief in branch C

LOI Risk Reduction Shown

- Risk reduction is the difference between the beginning and end scores for any two points in the lifecycle
  - This shows the change in confidence that a critical defect will leak to operations
  - The difference shows the result of IV&V activities in the given period in terms of reducing the opportunity for risk contributors to cause realization of the risk
  - This score can be computed periodically as analysis activities are completed to show a sequential risk reduction
  - A threshold could be defined to indicate the point of diminishing returns for effort on a given risk
- Items not yet in the method
  - The method does not incorporate size (e.g., function points) of the nodes. Having that would provide an expected number of critical defects leaking to operations which is a better representation of risk.
  - A defect present in operations does not necessarily mean the defect will be manifested. The method does not incorporate likelihood of risk being realized in expected operational scenarios.

Pilot Conclusions

- The method does provide a structured identification of a very specific set of IV&V activities to mitigate critical risks
- With consistency of estimation, relative reduction of risk can be stated quantitatively with a basis in actual data (with a Rayleigh curve extrapolation to operations)
- Not possible to say method definitely works
  - Actual risk at beginning and end cannot be determined absolutely
  - Assessment of risk is always subjective – ethereal concept
- Concurrence from a broad set of people with deep knowledge of the system is the best calibration source
  - Could set up a peer review process for scoring
- Results are consistent with what can be reasonably expected
- The risk level is per risk specific, it cannot be compared among other risks even within a project
- Improvements are available

Backup Slides

Risk Catalog Topical Areas

- Risks associated with
  - Software development principles
    - Requirement development
    - Architecture development
    - Design development
    - Code development
    - Test development and execution
  - System and software engineering principles
  - Hardware/software interface
  - Project management principles
  - Technology development principles
  - System resource management principles
  - Configuration management principles
  - Fault management and redundancy principles
  - System and software safety principles
  - System and software verification strategies
  - Operations
  - System modeling
  - Reuse strategies

Risk Catalog Features

- Risk characteristics
  - Applicable lifecycle phase
  - Typical criticality
  - Applicable development artifact
  - Applicability to one or more of the 3 questions
    - Does it affect normal operations
    - Does it affect handling of adverse conditions
    - Does it prevent inadvertent actions
  - Effective mitigation strategies
- Reasoning path
  - Hierarchical tiers of increasing detail
  - Thread found through response to questions
  - Incorporation of project unique risk sources
    - Fault management
    - Special technologies
    - Testing constraints
    - Hazard analyses
- Result is a guided search of extensive potential risk areas (from history and industry study) to provide comprehensive identification of project specific risks

Lightweight Spreadsheet Structure

Claim Levels						
1	2	3	4	5	Weight	Confidence
1 Claim	1.1 Claim	1.1.1 Claim	1.1.1.1 Claim	1.1.1.1.1 Claim	High	High
				1.1.1.1.2 Claim	High	Low
				1.1.1.1.3 Claim	Low	Low
			1.1.1.2 Claim	1.1.1.2.1 Claim	Low	Medium
				1.1.1.2.2 Claim	High	Medium
		1.1.2 Claim	1.1.2.1 Claim	1.1.2.1.1 Claim	High	Low
				1.1.2.1.2 Claim	High	High
				1.1.2.1.3 Claim	Low	Low
				1.1.2.1.4 Claim	Low	Low
			1.1.2.2 Claim	1.1.2.2.1 Claim	High	Low
				1.1.2.2.2 Claim	Low	Low
		1.1.3 Claim	1.1.3.1 Claim	1.1.3.1.1 Claim	Low	Low
				1.1.3.1.2 Claim	Low	Medium
			1.1.3.2 Claim	1.1.3.2.1 Claim	Low	Low
				1.1.3.2.2 Claim	Low	Low
	1.2 Claim	1.2.1 Claim	1.2.1.1 Claim	1.2.1.1.1 Claim	Medium	High
				1.2.1.1.2 Claim	Medium	Low
			1.2.1.2 Claim	1.2.1.2.1 Claim	Medium	Low
				1.2.1.2.2 Claim	Medium	Medium
				1.2.1.2.3 Claim	Medium	Low
		1.2.2 Claim	1.2.2.1 Claim	1.2.2.1.1 Claim	Medium	Low
				1.2.2.1.2 Claim	Medium	
			1.2.2.2 Claim	1.2.2.2.1 Claim	Medium	Low
				1.2.2.2.2 Claim	Medium	Low
2 Claim	2.1 Claim	2.1.1 Claim	2.1.1.1 Claim	2.1.1.1.1 Claim	High	Low
				2.1.1.1.2 Claim	High	Low
			2.1.1.2 Claim	2.1.1.2.1 Claim	High	Low
				2.1.1.2.2 Claim	Medium	Low
		2.1.2 Claim	2.1.2.1 Claim	2.1.2.1.1 Claim	Medium	Low
				2.1.2.1.2 Claim	High	High
			2.1.2.2 Claim	2.1.2.2.1 Claim	High	High
				2.1.2.2.2 Claim	High	High
	2.2 Claim	2.2.1 Claim	2.2.1.1 Claim	2.2.1.1.1 Claim	Medium	Medium
				2.2.1.1.2 Claim	Medium	Medium
			2.2.1.2 Claim	2.2.1.2.1 Claim	Medium	Medium
				2.2.1.2.2 Claim	Medium	Medium
		2.2.2 Claim	2.2.2.1 Claim	2.2.2.1.1 Claim	Medium	Low
				2.2.2.1.2 Claim	Medium	
			2.2.2.2 Claim	2.2.2.2.1 Claim	Medium	Low
				2.2.2.2.2 Claim	Medium	Low

Confidence	Weight	Color
High	High	Red
Medium	High	Red
Low	High	Green
High	Medium	Red
Medium	Medium	Yellow
Low	Medium	Green
High	Low	Yellow
Medium	Low	Green
Low	Low	Green
Unscored	High, Medium, Low	White

Importance Table

Contribution level	Characteristic Description	Value assigned
Risk realized	If this node has a known critical defect, the risk will definitely be realized	1.0
Risk close to realized	If one or two other associated nodes also have a known critical defect and FDIR is not present, the risk will definitely be realized	0.5
Risk probably realized	If one or two other associated nodes also have a known critical defect and FDIR is ineffective, the risk will definitely be realized	0.33
Risk could be realized	If one or two other disassociated nodes also have a known critical defect, the risk will definitely be realized	0.25
Risk may be realized	If two or more other disassociated nodes also have a known critical defect, the risk will definitely be realized	0.1
Risk not realized	If this node has no effect on risk realization (this node can probably be removed from the tree)	0

LOI Tree Snippet

2.1.1 VM LOI engine incorrectly specified	2.1.1.1 LOI VM does not has the highest priority	2.1.1.1.1 LOI sequence does not preempt lower priority sequences	MGSS-413-0046-SRD Virtual Machine Language Flight Component (VMLFC) v2.0.7 Software Requirements Document	Virtual Machine Language Flight Component (VMLFC) Version 2.0.9 Mission Planning and Sequencing Subsystem (SEQ) Virtual Machine Language (VML) Sequencing Release Description Document	Virtual Machine Language (VML) Version 2.0 User's Guide specifies which VM engine has priority (lowest number is highest priority)	MGSS DOC-0194, VML Flight Component (VMLFC) v2.0.9 Software Test Plan and Acceptance Test Report: 1)olvm_test verified correct off-line operation of vml environment including engine priorities 2)vml_flight_test verified proper operation of vml priority scheme 3) vml_compiler_test properly tested the vml engine priority specification
		2.1.1.1.2 LOI sequence preempted by lower priority sequences	Same as Requirement evidence for 2.1.1.1.1	Same as Design evidence for 2.1.1.1.1	Virtual Machine Language (VML) Version 2.0 User's Guide specifies which VM engine has priority (lowest number is highest priority)	MGSS DOC-0194, VML Flight Component (VMLFC) v2.0.9 Software Test Plan: vml_compiler_test properly tested the vml engine priority specification
	2.1.1.2 Multiple VM engines use common SC assets	2.1.1.2.1 LOI sequence denied access to needed SC assets	Same as Requirement evidence for 2.1.1.1.1	Same as Design evidence for 2.1.1.1.1	Virtual Machine Language (VML) Version 2.0 User's Guide specifies how to use SC assets along with GRAIL adaptation data files	MGSS DOC-0194, VML Flight Component (VMLFC) v2.0.9 Software Test Plan: adaptation_plan tested GRAIL specific designs.

LOI Evidence and Scores Snippet

Sub-claim level 3	Rolled score	Sub-claim level 4	Rolled score	Sub-claim level 5	Weight	Weight rationale	Sum of range scores	Requirements Range Total .203 - .007 Sev 1 & 2 .007 - .0002	Design Range .582 - .001 .02 - .000036	Code Range .275 - .01 .0099 - .00036
1.1.2 FSW does not control Propulsion Actuator hardware correctly.	0.0055468	1.1.2.1 High pressure latch valve is not commanded open correctly.	0.000596	The latch valve does not open when commanded to open.	1	If the valve cannot be opened, propulsion cannot be controlled	0.000596	0.0002	0.000036	0.00036
		1.1.2.2 High pressure latch valve is not commanded closed correctly.	0.000596	The latch valve does not close when commanded to close.	1	If the valve cannot be closed, propulsion cannot be controlled	0.000596	0.0002	0.000036	0.00036
		1.1.2.3 Pyro valve 1 (Helium)is not commanded correctly (fired).	0.000596	Pyro valve 1 valve does not open when commanded to open.	1	If the valve cannot be opened, propulsion cannot be controlled	0.000596	0.0002	0.000036	0.00036

FRR Tree Snippet

The Flight Readiness Review (FRR) is ineffective in establishing software readiness for flight	Review approach is inadequate	Participants don't understand the review process	
		Process for handling Review Item Discrepancies is not conducive to good review	no opportunities for reviewers to discuss issues with developers before RID submittal
			execution of the process is sloppy
			review board tendency to reject high percentage of issues
			review board or its subtiers inadequately staffed (too few or expertise to low)